PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 :
G06F 15/16, 15/163

A1

(11) International Publication Number: WO 00/58853

(43) International Publication Date: 5 October 2000 (05.10.00)

(21) International Application Number: PCT/US00/00739

(22) International Filing Date: 12 January 2000 (12.01.00)

(30) Priority Data:
09/283,077 31 March 1999 (31.03.99) US

(81) Designated States: AU, CA, JP, NZ, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

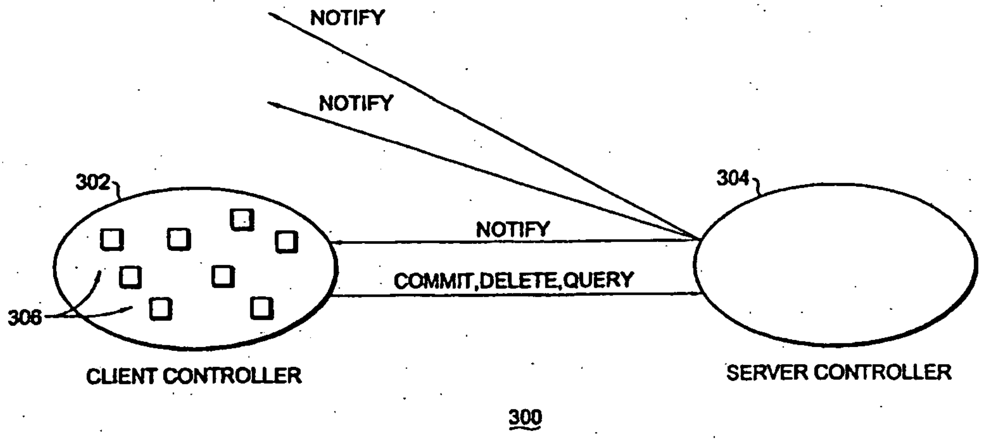
Published
With international search report.

(71) Applicant: CHANNELPOINT, INC. [US/US]; Suite 210, 10155 Westmoor Drive, Westminster, CO 80020 (US).

(72) Inventor: JACKSON, Jerry, R.; 11270 Wakely Road, Colorado Springs, CO 80908 (US).

(74) Agents: BURTON, Carol, W. et al.; Hogan & Hartson LLP, Suite 1500, 1200 17th Street, Denver, CO 80202 (US).

(54) Title: ADAPTIVE OPTIMIZATION OF CLIENT CACHING OF DISTRIBUTED OBJECTS



(57) Abstract

A system and method for adaptive optimization of client caching of distributed objects (306) in a client server computer system (300) is operational such that when a client system (302) requests multiple objects (306) from an associated server (304), a subset of the selected object attributes is initially transmitted to the client (302) and cached locally prior to display of certain ones of the object attributes for operation. If a user operation then requires the fetch of an additional attribute for one of the requested objects, it is either fetched from the client cache (302) if already cached, or automatically and transparently fetched from the server (304) for subsequent caching at the client (302), whereupon the additional attribute is then displayed. In those instances wherein a particular client (302) in the client-server computer system (300) commits a change to any of the object attributes, the associated server (304) then updates its own data and notifies any other clients and their applications that also have the particular object attribute to update their local cache as well.

AP2

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

ADAPTIVE OPTIMIZATION OF CLIENT CACHING OF DISTRIBUTED OBJECTS

BACKGROUND OF THE INVENTION

The present invention relates, in general, to the field of client-server computer systems. More particularly, the present invention relates to a system and method for adaptive optimization of client caching of distributed objects in a client server computer system.

Fundamentally, conventional distributed object computer systems have been implemented using two basic models. In a first system type, data and objects may reside on a first system which can then respond to messages to process data input to a second system to which it is coupled. An example of this technique in a client-server system configuration would be one in which the server maintains all data and objects and then conducts all processing at the client's request for presentation of only the results at the client. As is obvious, this is not a particularly effective methodology due to the fact that a great deal of time and system interconnect resources must be utilized to process any given request and then present the results at the requesting system, thereby unacceptably slowing overall system operation.

An alternative technique that ameliorates some of the problems inherent in the foregoing model is one in which the client system would initially pull all requested objects and their attributes from the server and then process the information at the client itself. However, when a potentially large number of objects are involved, the wait for the transfer of all of the objects and their attributes, and the attendant load this places on the client-server interconnect, memory and central processing unit, make this also an unacceptably slow solution in addition to a very clumsy programming model as the client will ultimately have to wait for all of the objects to be transferred before any operations can be performed on them. Such methodology also results in the filling of the local client cache with a great deal of data that is likely not to ever be needed.

SUMMARY OF THE INVENTION

Consequently, it would be most desirable to avoid either of these two methods for processing data in a client-server computer system. In accordance with the system and method of the present invention, when a client system requests multiple objects from an associated server, a subset of the selected object attributes is initially transmitted to the client and cached locally prior to display (or manipulation) of certain ones of the object attributes for operation. If a user operation then requires the fetch of an additional attribute for one of the requested objects, it is either fetched from the client cache if already cached, or automatically and transparently fetched from the server for subsequent caching at the client, whereupon the additional attribute is then displayed or manipulated. In those instances wherein a particular client in the client server computer system commits a change to any of the object attributes, the associated server then updates its own data and notifies any other clients and their applications that also have the particular object attribute to update their local cache as well.

In a particular exemplary application wherein, for example, a relatively large number of insurance providers is to be operated on at a client system, the selected attributes to be transmitted to the client from the server may only initially include their names and phone numbers, even though there are also a large number of other attributes for each insurance provider object. However, since these additional attributes are not initially required as the names and phone numbers are displayed for the user's initial operation or manipulation, they need not be transmitted to the client at this time obviating the wait and load on system interconnect, memory and central processing resources required to send them to the client until they are needed. As can be seen, utilizing the adaptive client caching technique of the present invention, computer programs may be implemented that, in essence, request a relatively large number of objects from a server, effectuate the processing of the object attributes at the client and not incur the system interconnect, memory and central processing load and delay penalties inherent in conventional client-server processing methodologies.

ChannelPoint, Inc., assignee of the present invention, has defined an object infrastructure for implementation of the system and method of the present invention in the form of a "huskobject". The server transmits these huskobjects to the client in response to a request for multiple objects as to which only certain attributes are initially of interest or otherwise specified by the client where they are then cached. The huskobjects do not contain all of the attributes of the objects but only those that are required for initial operation at the client. Should the client thereafter require an attribute not present in the huskobject and not found in the local cache, a request is automatically sent to the server to send the additional attribute. In operation, this request is transparent and it appears to the client that all of the attributes of the objects have previously been transmitted from the server. In this manner, interconnect traffic between the client and server is minimized.

Once the huskobjects are at the client, cache consistency with other clients is maintained by a notification mechanism from the server which is informed of a client object attribute change commit over the bi-directional data communication interconnect. When a data change is executed, the server updates its own data and also addresses the applications and caches on the other system clients which it knows have the same object attribute and would be affected by the change.

Particularly disclosed herein is a method, and computer program product for effectuating a method, for processing information in a client-server computer system. The method comprises the steps of initiating a request at a client for multiple objects from an associated server, transmitting selected attributes associated with each of the multiple objects from the server to the client and storing the selected attributes in a cache memory at the client. The selected attributes are stored at the client for user operation and at least one additional attribute required by user operation on one of the multiple objects is fetched from the cache memory if previously stored therein or alternatively fetched from the server and additionally stored in the cache memory.

Further disclosed herein is a client-server computer system which comprises a client computer having an associated cache memory and a server computer bi-directionally coupled to the client computer by means of a data communications path. The client computer is operational to request from the server computer at least one object having a number of attributes associated therewith and the server computer responsively transmits a selected subset of the number of attributes associated with the object to the client computer in response to the request. The client computer automatically fetches a required additional attribute for the object other than the selected subset from the associated cache memory or the server computer.

BRIEF DESCRIPTION OF THE DRAWINGS

The aforementioned and other features and objects of the present invention and the manner of attaining them will become more apparent and the invention itself will be best understood by reference to the following description of a preferred embodiment taken in conjunction with the accompanying drawings, wherein:

Fig. 1 illustrates an exemplary general distributed computing system for possible use in conjunction with the system, method and computer program product of the present invention wherein general purpose computers, workstations or personal computers are connected via communications links of various types;

Fig. 2 is a simplified representation of a client-server computer system illustrating a number of different objects and the relationship between client controllers and server controllers;

Fig. 3 is an additional simplified representation of the interrelationship between a client controller and server controller wherein a commit, delete or query operation is initiated by a given client controller and the associated server controller then notifies all affected other client controllers of that operation; and

Fig. 4 is a representative logic flowchart of a method in accordance with a particular implementation of the present invention.

DESCRIPTION OF A PREFERRED EMBODIMENT

With reference now to Fig. 1, the environment in which the present invention may be used is illustrated as encompassing a general distributed computing system 10 wherein general purpose computers, workstations or personal computers are connected via communications links of various types, for example a client-server arrangement, and wherein programs and data, many in the form of objects, are made available by various members of the system 10 for execution and access by other members of the system. Some of the elements of a representative computer system 10 are shown including a general purpose workstation 12 (which may function as a "client" in a client-server computer system 10) and associated server 14 coupled together through an appropriate communications medium 16.

The workstation 12 may include input/output ("I/O"), central processing unit ("CPU") and memory sections (not shown) forming a portion of a computer 18 having an associated monitor 20. A keyboard 22 as well as other manual input devices, such as mouse 24, form a portion of the workstation 12 and are coupled to the I/O section of the computer 18 to provide user input data thereto. The monitor 20 is coupled to receive output from the I/O section to provide visually discernible user output in the form of a graphical user interface ("GUI") or other form of user interface. Various types of computer mass storage devices may also be coupled to the I/O section of the computer 18 including a tape drive 26, a hard disk drive 28, a floppy disk drive 30 or a CDROM drive 32 for read/write or read only storage of data.

Data may also be communicated to the server 14 by means of the communications medium 16 (such as a local area network "LAN" or wide area network "WAN") intercoupling the I/O portions of the workstation 12 and server 14. The server 14 may comprise, for example, computer tower 34, an associated monitor 36, a keyboard 38 and computer mass storage device 40. The computer program products containing mechanisms to effectuate the apparatus and methods of the present invention may reside in the memory portions of the workstation 12 or server 14 or any of the various associated computer mass storage devices such as tape drive 26, hard disk drive 28, floppy disk drive 30, CDROM drive 32, the communications medium 16, the computer mass storage device 40 or any other device or mechanism for retaining computer program data.

With reference additionally now to Fig. 2, a simplified representation of a client-server computer system 200 is shown. The computer system 200 includes a number of client computers 202, 204 on which are running various application programs denominated Application 1 and Application 2 respectively. A Server Process is illustrated as running on server computer 206. In this figure, a number of different objects (for example, "MCC", "PCC" and "DCC") on the client computers 202, 204 are shown (where "CC" = Client Controller) and their relationship to the corresponding objects ("MSC", "PSC" and "DSC") on the server 206 (where "SC" = Server Controller).

With reference additionally now to Fig. 3, an additional simplified representation of a client-server computer system 300 is shown illustrating the interrelationship between a client controller 302 and a server controller 304. When a commit, delete or query operation is initiated by a given client (for example, client controller 302) with respect to one of a number of huskobjects 306, knowledge of this operation is then passed to the server controller 304. The associated server controller 304 then modifies its own corresponding data accordingly and further notifies all other affected client controllers of that operation such that the change can be acted upon by their application program and any new value of the attribute added to their respective cache.

The query from the client controller 302 to the server 304 is used to specify the objects desired and the attributes associated with those objects that are requested to be transmitted because of their immediate interest to the client application.

With reference additionally now to Fig. 4, a representative process 400 in accordance with the present invention is shown. The process 400 is initiated when a given client in a client-server computer system requests multiple objects from an associated server at step 402. At step 404, the server responsively transmits selected attributes of the requested objects to the client instead of transmitting all attributes associated with the requested objects. When the requested objects with their selected attributes are received at the client, they are cached locally at step 406 and then displayed (or manipulated) for user operation at step 408.

When a user operation at the client causes the need to fetch at least an additional attribute of any object at step 410, a determination is made at decision step 412 as to whether the additional attribute is already cached locally at the client location. If it is, then at step 414, the additional attribute is fetched from the local cache. Alternatively, if the additional object has not previously been cached at the client, the client automatically and transparently fetches the additional object from the associated server at step 416. Thereafter, whether fetched from the local cache or from the server, the additional attribute is now available at the client at step 420.

In the event that the particular client system commits a change to any locally cached attribute at decision step 422, the server is made aware of this and then proceeds to update its own data as well as notify any other affected clients and their applications to update the data and their local cache at step 424.

An exemplary illustration of the system and method of the present invention is for a client user to request all insurance providers (i.e. objects) in a given geographic location. In response to receipt of the object data, the client system will display or manipulate them, and from that displayed list, select the one of interest. As can be seen, since it is highly likely that the user will really only need to see a list of, perhaps, the name and address of the various insurance providers at first, huskobjects containing only the name and address of the insurance providers requested is really all that is needed initially. To have provided all available attributes for each of the providers would not only have unacceptably increased the client-server data traffic and resulted in a long wait for the data to be transferred, it would also serve to over-populate the client's cache with many attributes and objects which will never be needed.

Through the use of the system and method of the present invention, objects can be dealt with at the client location as if the objects and their attributes were completely present without having to transmit everything from the server in bulk. Those attributes that are initially requested may be set by the application software based on the likelihood that certain attributes of the objects are the ones most likely to be needed at first and those associated with them that is most likely to be needed next once an object of interest has been determined. The application software can also determine that if certain attributes (e.g. x, y and z) are initially requested, then there is a high probability that attribute "q" will be needed later. In this example, attributes "x", "y', "z" and "q" may be sent at one time. In operation, objects and their attributes can be fetched by following links to other objects or by query which specifies the attributes of particular interest when a number of objects are requested from the server. In other instances, attributes can be pinned to each other such that, for example, a request for attribute "x" would also always bring over "y" if the latter where pinned to the former.

While there have been described above the principles of the present invention in conjunction with specific client-server computer system configurations and object infrastructure it is to be clearly understood that the foregoing description is made only by way of example and not as a limitation to the scope of the invention. Particularly, it is recognized that the teachings of the foregoing disclosure will suggest other modifications to those persons skilled in the relevant art. Such modifications may involve other features which are already known per se and which may be used instead of or in addition to features already described herein. Although claims have been formulated in this application to particular combinations of features, it should be understood that the scope of the disclosure herein also includes any novel feature or any novel combination of features disclosed either explicitly or implicitly or any generalization or modification thereof which would be apparent to persons skilled in the relevant art, whether or not such relates to the same invention as presently claimed in any claim and whether or not it mitigates any or all of the same technical problems as confronted by the present invention. The applicants hereby reserve the right to formulate new claims to such features and/or combinations of such features during the prosecution of the present application or of any further application derived therefrom.

What is claimed is:

CLAIMS:

1. A method for processing information in a client-server computer system comprising:

initiating a request at a client for multiple objects from an associated server;

transmitting selected attributes associated with each of said multiple objects from said server to said client;

storing said selected attributes in a cache memory at said client;

displaying said selected attributes at said client for user operation;

fetching at least one additional attribute required by said user operation on one of said multiple objects from said cache memory if previously stored therein; and

alternatively fetching said at least one additional attribute from said server and additionally storing said at least one additional attribute in said cache memory.

2. The method of claim 1 wherein said steps of alternatively fetching is carried out automatically in response to a determination at said client that said at least one additional attribute is not previously stored in said cache memory.

3. The method of claim 1 further comprising the step of:

also displaying said additional attribute at said client.

4. The method of claim 1 further comprising the step of:

committing a change at said client to one of said attributes associated with one of said multiple objects; and

communicating said committed change to said server.

5. The method of claim 4 further comprising the step of:

updating data at said server associated with said change to one of said attributes associated with one of said multiple objects.

6. The method of claim 5 further comprising the step of:

notifying other clients associated with said server affected by said change to one of said attributes associated with one of said multiple objects.

7. The method of claim 6 further comprising the step of:

storing said change to one of said attributes associated with one of said multiple objects to a cache memory at said other affected clients.

8. A client-server computer system comprising:

a client computer having an associated cache memory;

a server computer bi-directionally coupled to said client computer by means of a data communications path;

wherein said client computer is operational to request from said server computer at least one object having a number of attributes associated therewith, said server computer responsively transmitting a selected subset of said number of attributes associated with said object to said client computer in response to said request and said client computer automatically fetching a required additional attribute for said object other than said selected subset from said associated cache memory or said server computer.

9. The client-server computer system of claim 8 wherein said client computer further comprises a display for displaying said selected subset of said number of attributes associated with said object.

10. The client-server computer system of claim 8 wherein said client computer automatically fetches required additional attribute for said object from said server computer if said required additional attribute is not already in said associated cache memory.

11. The client-server computer system of claim 8 wherein said data communications path comprises a computer network.

12. The client-server computer system of claim 8 wherein a change to an attribute of said object committed at said client computer is

communicated to said server by means of said data communications path.

13. The client-server computer system of claim 12 wherein said server computer is operational to update a corresponding attribute in response to said change committed at said client computer.

14. The client-server computer system of claim 13 further comprising:

an additional client computer having an associated cache memory associated with said server computer by another data communications path.

15. The client-server computer system of claim 14 wherein said server computer is operational to notify said additional client computer of said change committed at said client computer if said additional client computer has said corresponding attribute stored in its associated cache memory.

16. A computer program product comprising:

a computer usable medium having computer readable code embodied therein for processing information in a client-server computer system comprising:

computer readable program code devices configured to cause a computer to effect initiating a request at a client for multiple objects from an associated server;

computer readable program code devices configured to cause a computer to effect transmitting selected attributes associated with each of said multiple objects from said server to said client;

computer readable program code devices configured to cause a computer to effect storing said selected attributes in a cache memory at said client;

computer readable program code devices configured to cause a computer to effect displaying said selected attributes at said client for user operation;

computer readable program code devices configured to cause a computer to effect fetching at least one additional attribute required by said user operation on one of said multiple objects from said cache memory if previously stored therein; and

computer readable program code devices configured to cause a computer to effect alternatively fetching said at least one additional attribute from said server and additionally storing said at least one additional attribute in said cache memory.

17. The computer program product of claim 16 wherein said computer readable program code devices configured to cause a computer to effect alternatively fetching is carried out automatically in response to a determination at said client that said at least one additional attribute is not previously stored in said cache memory.

18. The computer program product of claim 16 further comprising:

computer readable program code devices configured to cause a computer to effect also displaying said additional attribute at said client.

19. The computer program product of claim 16 further comprising:

computer readable program code devices configured to cause a computer to effect committing a change at said client to one of said attributes associated with one of said multiple objects; and

computer readable program code devices configured to cause a computer to effect communicating said committed change to said server.

20. The computer program product of claim 19 further comprising:

computer readable program code devices configured to cause a computer to effect updating data at said server associated with said change to one of said attributes associated with one of said multiple objects.

21. The computer program product of claim 20 further comprising:

computer readable program code devices configured to cause a computer to effect notifying other clients associated with said server affected by said change to one of said attributes associated with one of said multiple objects.

22. The computer program product of claim 21 further comprising:

computer readable program code devices configured to cause a computer to effect storing said change to one of said attributes associated with one of said multiple objects to a cache memory at said other affected clients.

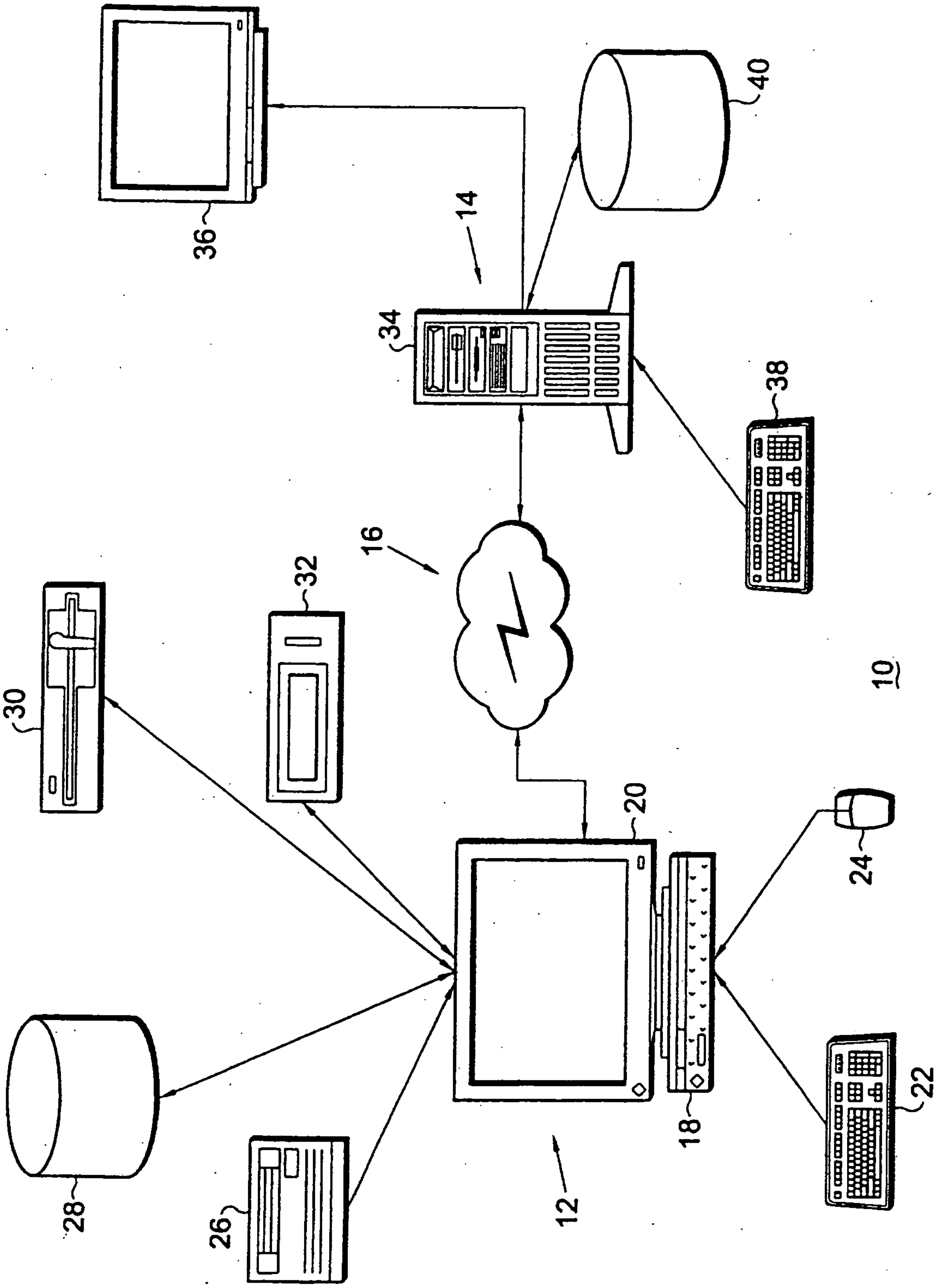


FIG. 1

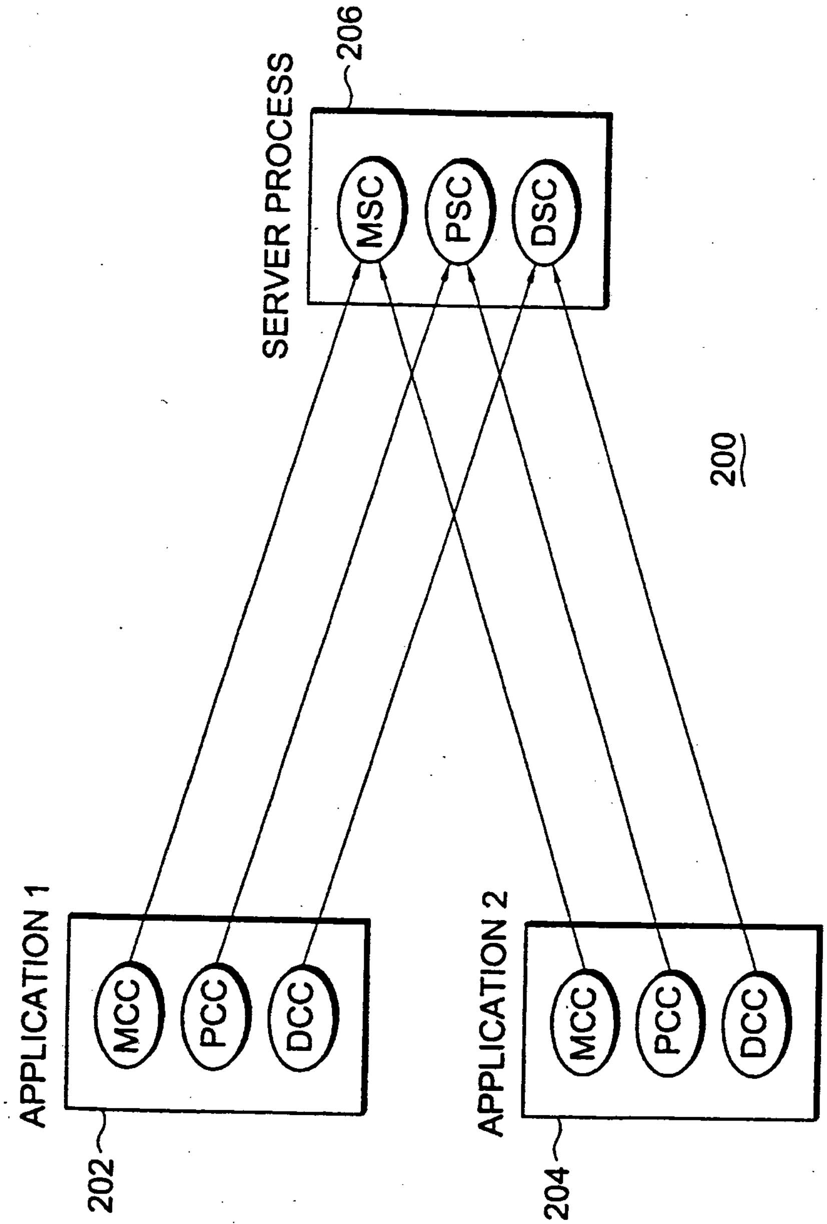

SERVER PROCESS
206
MSC
PSC
DSC
200
APPLICATION 1
202
MCC
PCC
DCC
APPLICATION 2
204
MCC
PCC
DCC

FIG. 2

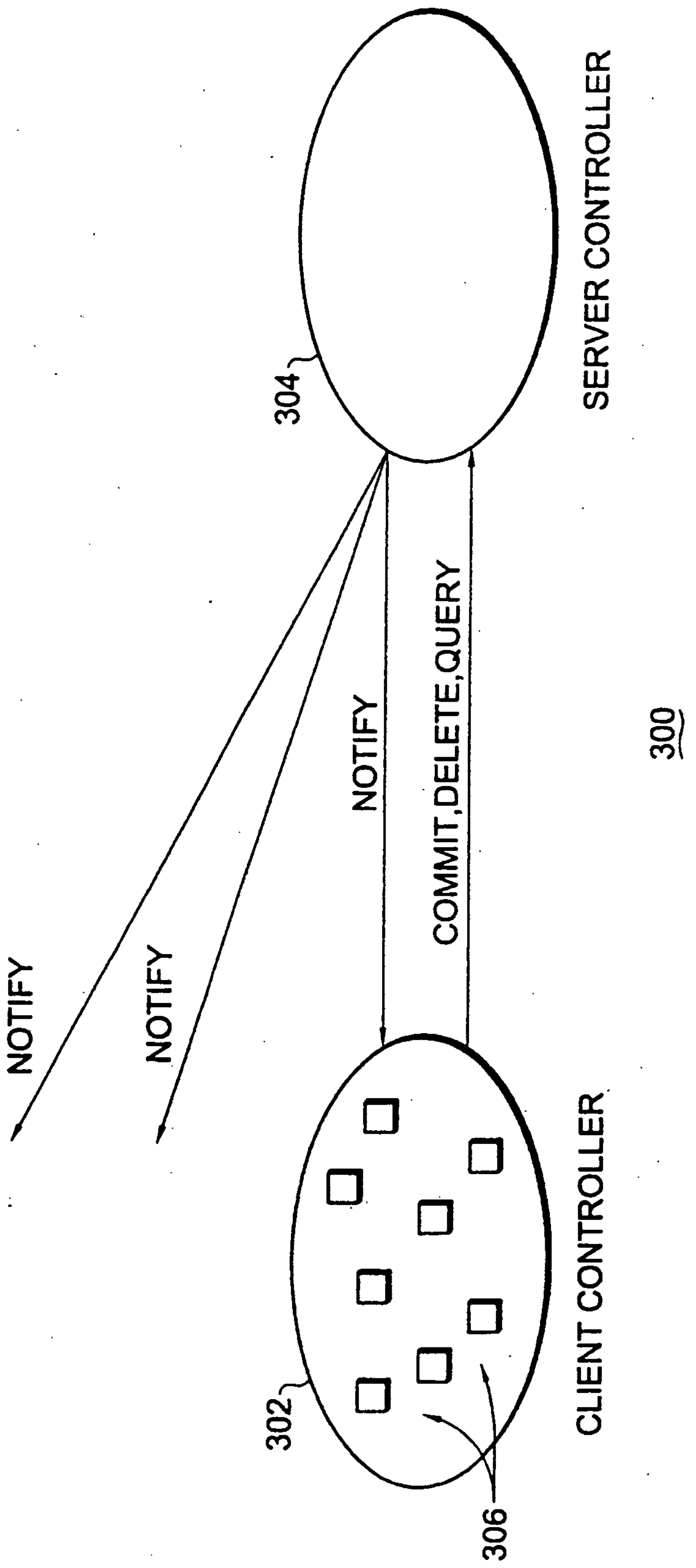

SERVER CONTROLLER
304
NOTIFY
COMMIT,DELETE,QUERY
NOTIFY
NOTIFY
300
CLIENT CONTROLLER
302
306

FIG. 3

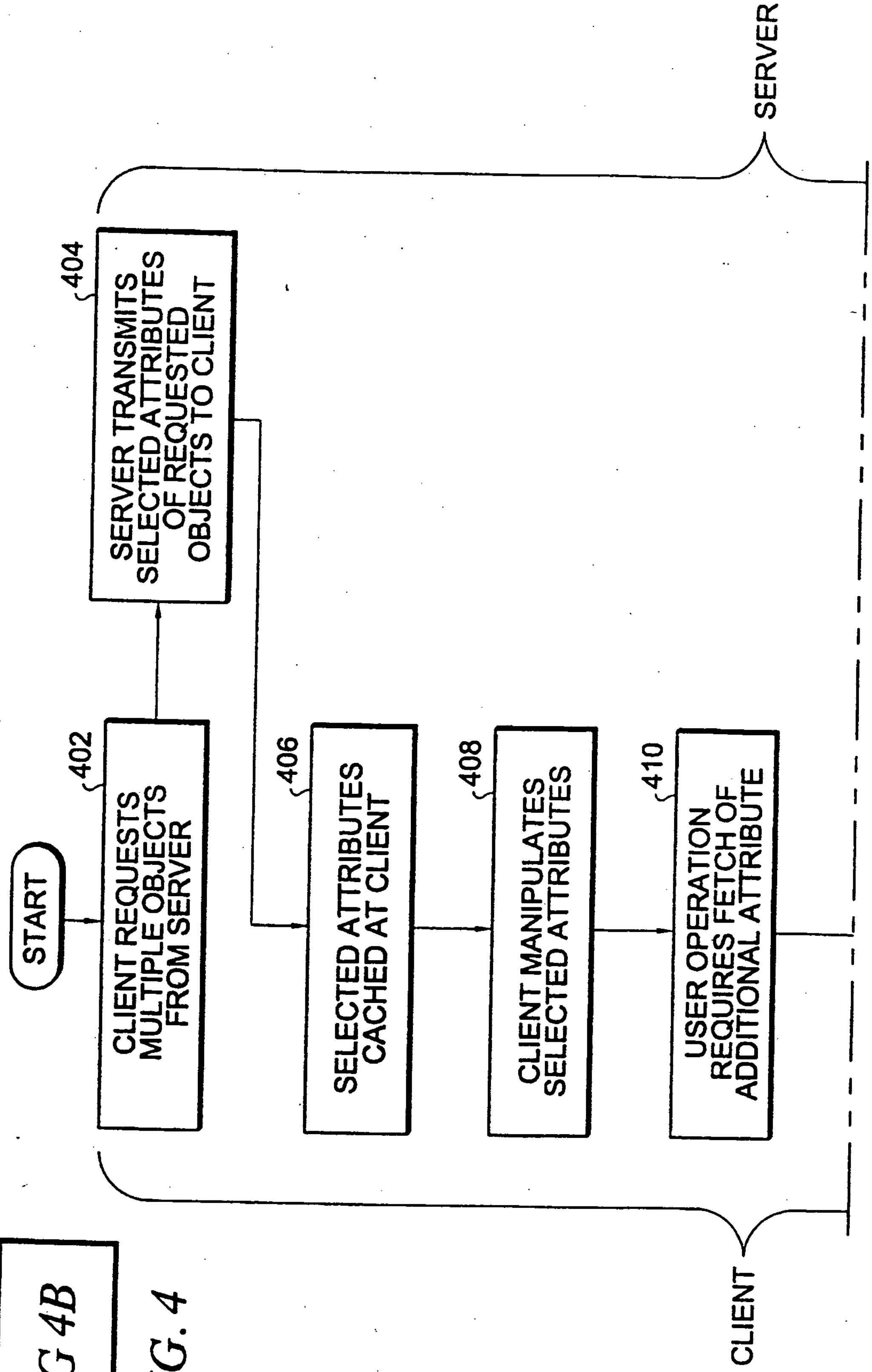

SERVER
404
SERVER TRANSMITS SELECTED ATTRIBUTES OF REQUESTED OBJECTS TO CLIENT
START
402
CLIENT REQUESTS MULTIPLE OBJECTS FROM SERVER
406
SELECTED ATTRIBUTES CACHED AT CLIENT
408
CLIENT MANIPULATES SELECTED ATTRIBUTES
410
USER OPERATION REQUIRES FETCH OF ADDITIONAL ATTRIBUTE
CLIENT

FIG 4A

FIG 4A
FIG 4B

FIG. 4

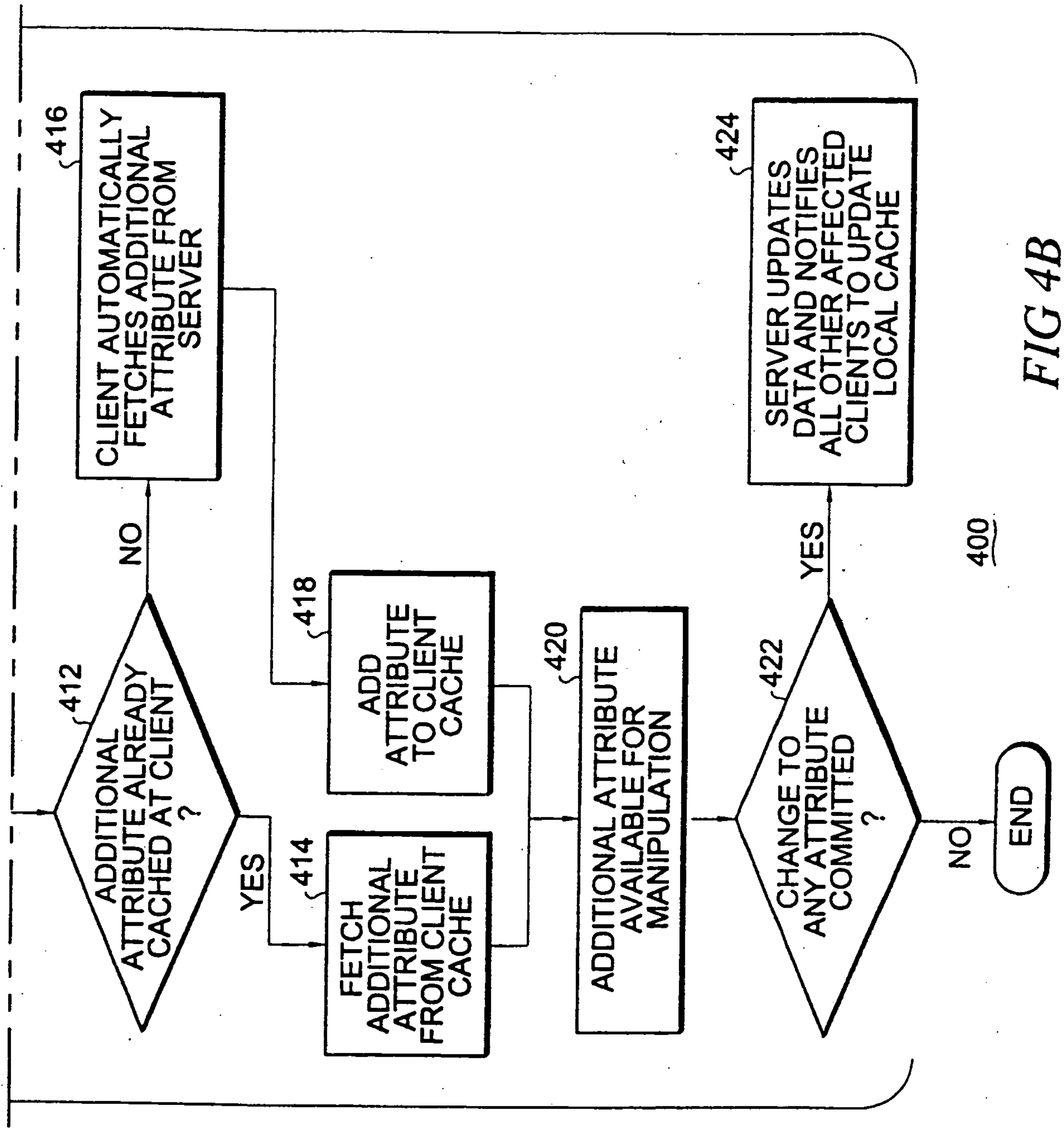

412
ADDITIONAL ATTRIBUTE ALREADY CACHED AT CLIENT ?
YES
NO
414
FETCH ADDITIONAL ATTRIBUTE FROM CLIENT CACHE
416
CLIENT AUTOMATICALLY FETCHES ADDITIONAL ATTRIBUTE FROM SERVER
418
ADD ATTRIBUTE TO CLIENT CACHE
420
ADDITIONAL ATTRIBUTE AVAILABLE FOR MANIPULATION
422
CHANGE TO ANY ATTRIBUTE COMMITTED ?
NO
END
YES
424
SERVER UPDATES DATA AND NOTIFIES ALL OTHER AFFECTED CLIENTS TO UPDATE LOCAL CACHE
400

FIG 4B

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/00739

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/16, 15/163
US CL :709/203, 310
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/203, 310

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST, EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,737,536 A (HERRMANN ET AL) 07 APRIL 1998, (COL. 5, LINE 57-COL. 6, LINE 20; COL. 15, LINES 7-16; COL. 19, LINES 39-53)	1-22
Y, E	US 6,021,470 A (FRANK ET AL) 01 FEBRUARY 2000, ALL	1-22
Y	US 5,884,046 A (ANTONOV) 16 MARCH 1999, (COL. 6, LINES 17-63)	1-22
A	US 5,925,100 A (DREWRY ET AL) 20 JULY 1999	1-22
A	US 5,842,216 A (ANDERSON ET AL) 24 NOVEMBER 1998	4-7, 12-15, 19-22

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier document published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search 14 MARCH 2000	Date of mailing of the international search report 15 JUN 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer ALVIN OBERLEY
Facsimile No. (703) 305-3230	Telephone No. (703) 305-9716